Paper Number: IN11C-1061

NATIONAL AERONAUTICS
AND SPACE ADMINISTRATION
ESDS Reuse Working Group

Measuring Software Reusability for Scientific Data Systems

Background

Reusing existing software components and related artifacts offers the potential to reduce costs and to improve the quality of systems and applications that support Earth science and other scientific disciplines. Capabilities to measure the reusability of software artifacts can have benefits for developers and adopters of information systems, but reusability is generally omitted from most measurements of technology readiness. Therefore, the NASA Earth Science Data Systems Software Reuse Working Group is developing Reuse Readiness Levels (RRLs) to facilitate the measurement of software reusability.

The nine topic areas considered by the Working Group (WG) during the development of the RRLs are:

- Documentation
- Extensibility
- Intellectual Property Issues
- Modularity
- Packaging
- Portability
- Standards Compliance
- Support
- Verification/Testing

The WG follows an iterative process to develop the RRLs, first writing levels for the topic areas, then looking across all topic areas at each level to create the RRLs, and making suggested revisions to RRLs and topic area levels based on community feedback. Also, the WG is developing a set of use cases for the RRLs, and plans to assess some existing reusable assets to test the consistency of the RRLs.

Draft Reuse Readiness Levels (RRLs)

Level	Summary	Description
RRL 1	Limited reusability; the software is not recommended for reuse.	Little is provided beyond limited source code or pre-compiled, executable binaries. There is no support, contact information for developers or rights for reuse specified, the software is not extensible, and there is inadequate or no documentation.
RRL 2	Initial reusability; software reuse is not practical.	Some source code, documentation, and contact information are provided, but these are still very limited. Initial testing has been done, but reuse rights are still unclear. Reuse would be challenging and cost-prohibitive.
RRL 3	Basic reusability; the software might be reusable by skilled users at substantial effort, cost, and risk.	Software has some modularity and standards compliance, some support is provided by developers, and detailed installation instructions are available, but rights are unspecified. An expert may be able to reuse the software, but general users would not.
RRL 4	Reuse is possible; the software might be reused by most users with some effort, cost, and risk.	Software and documentation are complete and understandable. Software has been demonstrated in a lab on one or more specific platforms, infrequent patches are available, and intellectual property issues would need to be negotiated. Reuse is possible, but may be difficult.
RRL 5	Reuse is practical; the software could be reused by most users with reasonable cost and risk.	Software is moderately portable, modular, extendable, and configurable, has low-fidelity standards compliance, a user manual, and has been tested in a lab. A user community exists, but may be a small community of experts. Developers may be contacted to request limited rights for reuse.
RRL 6	Software is reusable; the software can be reused by most users although there may be some cost and risk.	Software has been designed for extensibility, modularity, and portability, but software and documentation may still have limited applicability. Tutorials are available, and the software has been demonstrated in a relevant context. Developers may be contacted to obtain formal statements on restricted rights or to negotiate additional rights.
RRL 7	Software is highly reusable; the software can be reused by most users with minimum cost and risk.	Software is highly portable and modular, has high-fidelity standards compliance, provides auto-build installation, and has been tested in a relevant context. Support is developer-organized, and an interface guide is available. Software and documentation are applicable for most systems. Brief statements are available describing limited rights for reuse and developers may be contacted to negotiate additional rights.
RRL 8	Demonstrated local reusability; the software has been reused by multiple users.	Software has been shown to be extensible, and has been qualified through test and demonstration. An extension guide and organization-provided support are available. Brief statements are available describing unrestricted rights for reuse and developers may be contacted to obtain formal rights statements.
RRL 9	Demonstrated extensive reusability; the software is being reused by many classes of users over a wide range of systems.	Software is fully portable and modular, with all appropriate documentation and standards compliance, encapsulated packaging, a GUI installer, and a large support community that provides patches. Software has been tested and validated through successful use of application output. Multiple statements describing unrestricted rights for reuse and the recommended citation are embedded into the product.

Potential users and uses for the RRLs include:

- Adopters of software and related artifacts
  - RRLs could serve as metadata for reusable software assets stored in catalogs and repositories to provide guidance and enable evaluation of the potential reusability of software assets and system components being considered for adoption.
- Developers of systems and software applications
  - RRLs could serve as an indicator to provide guidance on areas needing further development when creating reusable assets.
- Sponsors of projects involving software development
  - RRLs could serve as parts of requests for proposals or contracts, asking for a reuse approach or how assets are being made reusable.

Potential tools involving the RRLs include:

- RRL Calculator, which can calculate an overall RRL from a set of topic area levels
- RRL Assessment Guidelines, which can help users assess reusable software assets
- Reuse Readiness Improvement Guidelines, which can help developers increase the reuse readiness of assets they develop

Authors:

James J. Marshall*, Innovim / NASA GSFC
(James.J.Marshall@nasa.gov)
Robert R. Downs, CIESIN, Columbia University
Ryan S. Gerard**, Innovim / NASA GSFC

* Contact for WG information
** Presenter

For more information, please visit **http://www.esdswg.com/softwarereuse**